Parameter-robust preconditioning for the optimal control of the wave equation

Parameter-Robust Preconditioning for the Optimal Control of the Wave Equation ☆

Jun Liu[a], John W. Pearson[b]

[a]*Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA.*
[b]*School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Edinburgh, EH9 3FD, UK.*

Abstract

In this paper, we propose and analyze a new matching-type Schur complement preconditioner for solving the discretized first-order necessary optimality conditions that characterize the optimal control of wave equations. Coupled with this is a recently developed second-order implicit finite difference scheme used for the full space-time discretization of the optimality system of PDEs. Eigenvalue bounds for the preconditioned system are derived, which provide insights into the convergence rates of the preconditioned Krylov subspace method applied. Numerical examples are presented to validate our theoretical analysis and demonstrate the effectiveness of the proposed preconditioner, in particular its robustness with respect to very small regularization parameters, and all mesh-sizes in the spatial variables.

Keywords: Preconditioning, Optimal control, Wave equation, Finite difference method, Schur complement, Regularization, Saddle-point system

1. Introduction

Optimal control problems governed by time-dependent partial differential equations (PDEs) [1, 2, 3, 4, 5] have captured much attention from the scientific computing and wider engineering communities in the last few decades, due to their substantial utility, and the computational challenges associated with them. Such PDE control problems appear in a wide range of applications such as flow control design [6], aerodynamic shape optimization [7], and photoacoustic tomography [8], to name but a few. In this paper, we propose and analyze a new matching-type Schur complement preconditioner for iteratively solving the large-scale discretized optimality systems arising from optimal control problems governed by the wave equation. The solution of such optimization problems involving hyperbolic PDEs are especially difficult due to the Helmholtz-type operators which often need to be tackled upon discretization. To date, the development of preconditioners for optimal control problems has largely focused on elliptic and parabolic problems which do not face these issues (see [9, 10, 11, 12, 13, 14, 15, 16, 17], for example). In this paper, we attempt to bridge part of the gap between the existing preconditioning theory, and the solution of a class of optimization problems constrained by a hyperbolic system.

Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, be a spatial domain with boundary $\partial\Omega$. The results presented in this paper hold for any choice of d, though from an analytical point of view we will largely focus on the standard case

☆Jun Liu's research was supported by a "Seed Grants for Transitional and Exploratory Projects" (STEP) Award (FY2019) from the SIUE Graduate School. John W. Pearson's research was supported by the Engineering and Physical Research Council (EPSRC) Fellowship EP/M018857/2, and a Fellowship of The Alan Turing Institute in London.

Email addresses: juliu@siue.edu (Jun Liu), j.pearson@ed.ac.uk (John W. Pearson)

$d = 2$. Given a finite period of time $T > 0$, we define $Q := \Omega \times (0, T)$ and $\Sigma := \partial\Omega \times (0, T)$. We consider the following distributed optimal control problem [4] of minimizing a tracking-type quadratic cost functional

$$J(y, u) = \tfrac{1}{2}\|y - g\|^2_{L^2(Q)} + \tfrac{\gamma}{2}\|u\|^2_{L^2(Q)} \tag{1}$$

subject to the linear wave equation along with initial and boundary conditions:

$$\begin{cases} y_{tt} - \Delta y = f + u & \text{in } Q, \\ y = 0 & \text{on } \Sigma, \\ y(\cdot, 0) = y_0 & \text{in } \Omega, \\ y_t(\cdot, 0) = y_1 & \text{in } \Omega, \end{cases} \tag{2}$$

where $u \in U := L^2(Q)$ is the distributed control function, $g \in L^2(Q)$ is the desired tracking trajectory, $\gamma > 0$ represents the weight of the cost of control (also defined as the Tikhonov regularization parameter), $f \in L^2(Q)$, and the initial conditions satisfy $y_0 \in H^1_0(\Omega)$ and $y_1 \in L^2(\Omega)$. The existence, uniqueness, and regularity of the solution for the optimal control problem (1)–(2) are well established [4]. By defining an appropriate Lagrange functional and making use of the strict convexity of $J(y, u)$, the optimal solution pair (y, u) of (1)–(2) is shown to be completely characterized by the unique solution pair (y, p) to the following first-order necessary optimality system:

$$\begin{cases} y_{tt} - \Delta y - \frac{1}{\gamma}p = f & \text{in } Q, \quad y = 0 \quad \text{on } \Sigma, \\ y(\cdot, 0) = y_0 & \text{in } \Omega, \quad y_t(\cdot, 0) = y_1 \quad \text{in } \Omega, \\ p_{tt} - \Delta p + y = g & \text{in } Q, \quad p = 0 \quad \text{on } \Sigma, \\ p(\cdot, T) = 0 & \text{in } \Omega, \quad p_t(\cdot, T) = 0 \quad \text{in } \Omega, \end{cases} \tag{3}$$

where the state y evolves forward in time and the adjoint state p marches backward in time. Here we have eliminated u by making use of the optimality condition $\gamma u - p = 0$. It is well-known that a key challenge for solving (3) results from the fact that the state y and the adjoint state p are marching in opposite orientations (due to the initial conditions for y, and the final-time conditions for p), meaning it is difficult to break up the numerical solution procedure into smaller sub-problems. The full numerical discretization of (3) leads to a large-scale coupled algebraic system of equations, since all time steps are resolved simultaneously [18], although it is often handled by constructing a fixed point iteration that consists of a forward and a backward solution step. However, such a forward-backward fixed point iteration is not guaranteed to result in a contraction, especially for very small values of γ (similar to the observations in [19, Theorem 4.1] and [20], for example).

In recent work [21], a new second-order accurate implicit finite difference scheme in both time and space for solving (3) was developed, which does not require any Courant–Friedrichs–Lewy (CFL) condition on the corresponding grid step-size ratio. Moreover, the resulting well-structured indefinite saddle-point system on the discrete level allowed the authors to design a fast iterative solver with a constraint preconditioner, which was shown to achieve a mesh-independent convergence rate. However, as also highlighted in [21], this constraint preconditioner exhibits a significantly slower convergence rate as the regularization parameter γ tends to zero, i.e., the number of iterations required to achieve convergence significantly increases when γ becomes smaller. Motivated by recent work [11, 22, 23, 13, 24, 25, 26] on regularization-robust preconditioners, our main contribution of this paper is to propose and analyze an improved preconditioner, based on a matching Schur complement strategy, that gives more robust convergence rates with respect to the

regularization parameter. In particular, some preliminary numerical results of [27] on the optimal control of the wave equation are now better supported by the eigenvalue analysis presented in this paper.

The paper is organized as follows. In Section 2 we present the full discretized linear system based on a recently developed implicit finite difference scheme, and discuss an established constraint preconditioner based on its known eigenvalue bounds. A new matching-type Schur complement preconditioner is proposed and analyzed in Section 3, where eigenvalue bounds of the preconditioned system are obtained. Numerical experiments are performed in Section 4 to validate our theoretical analysis and to demonstrate the effectiveness of the proposed preconditioner. Finally, concluding remarks are made in Section 5.

2. Discretized Optimality System and a Constraint Preconditioner

In the subsequent finite difference approach, we follow the approach of [21], where the two-dimensional domain $\Omega = (0,1)^2$ is considered (although of course much more general domains can also be handled by the finite difference method). We discretize the system (3) using the leap-frog scheme in time with an averaged five-point second-order central difference discretization in space. More specifically, we partition the time interval $[0,T]$ uniformly into $0 = t_0 < t_1 < \cdots < t_N = T$ with $t_k - t_{k-1} = \tau = \frac{T}{N}$, and discretize the spatial domain Ω uniformly into $0 = \xi_0 < \xi_1 < \cdots < \xi_{M_1} = 1$ and $0 = \zeta_0 < \zeta_1 < \cdots < \zeta_{M_2} = 1$, with $h_1 = \xi_i - \xi_{i-1}$, $h_2 = \zeta_j - \zeta_{j-1}$. We denote $h = \max\{h_1, h_2\}$. We also define the discrete Laplacian

$$(\Delta_h Y^n)_{ij} = \frac{Y^n_{i-1,j} - 2Y^n_{i,j} + Y^n_{i+1,j}}{h_1^2} + \frac{Y^n_{i,j-1} - 2Y^n_{i,j} + Y^n_{i,j+1}}{h_2^2}$$

in the $d = 2$ case, and analogously for the $d = 1$ and $d = 3$ cases.

Let I be an identity matrix of appropriate size, the vectors y_0 and y_1 denote the lexicographic ordering (vectorization) of the initial conditions over spatial grid points, and the vectors f^n, g^n, y^n, and p^n correspond to the lexicographic ordering of the corresponding function approximations over spatial grid points at the n-th time step. Upon lexicographic ordering of the unknown approximations, the implicit finite difference scheme developed in [21] can be formulated as a symmetric indefinite saddle-point system

$$A_h \begin{bmatrix} y_h \\ p_h \end{bmatrix} := \begin{bmatrix} \check{I}_h & L_h^\intercal \\ L_h & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix} \begin{bmatrix} y_h \\ p_h \end{bmatrix} = \begin{bmatrix} g_h \\ f_h \end{bmatrix}, \tag{4}$$

where

$$L_h = \frac{1}{\tau^2} \begin{bmatrix} D_h & 0 & 0 & 0 & \cdots & 0 \\ -2I & D_h & 0 & 0 & \cdots & 0 \\ D_h & -2I & D_h & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & D_h & -2I & D_h & 0 \\ 0 & 0 & \cdots & D_h & -2I & D_h \end{bmatrix} \quad \text{with} \quad D_h = I - \frac{\tau^2}{2}\Delta_h, \tag{5}$$

$$\hat{I}_h = \begin{bmatrix} \frac{1}{2}I & 0 & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & 0 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I & 0 \\ 0 & 0 & \cdots & 0 & 0 & I \end{bmatrix}, \quad \check{I}_h = \begin{bmatrix} I & 0 & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & 0 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I & 0 \\ 0 & 0 & \cdots & 0 & 0 & \frac{1}{2}I \end{bmatrix},$$

$$f_h = \begin{bmatrix} \frac{1}{2}f^0 + \frac{1}{\tau}y_1 + \frac{1}{\tau^2}y_0 \\ f^1 - \frac{1}{\tau^2}D_h y_0 \\ f^2 \\ \vdots \\ f^{N-2} \\ f^{N-1} \end{bmatrix}, \quad g_h = \begin{bmatrix} g^1 \\ g^2 \\ \vdots \\ g^{N-1} \\ \frac{1}{2}g^N \end{bmatrix}, \quad y_h = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{N-1} \\ y^N \end{bmatrix}, \quad p_h = \begin{bmatrix} p^0 \\ p^1 \\ p^2 \\ \vdots \\ p^{N-1} \end{bmatrix}.$$

Note that this discretization guarantees the presence of symmetric positive definite matrices D_h on the diagonal blocks of the block upper triangular matrix L_h, which will be crucial when constructing preconditioners for the matrix system. Here $L_h^\intercal$ denotes the transpose of the matrix L_h.

For such two-by-two block sparse saddle-point systems, a range of numerical algorithms have been summarized in [28]. However, many of the existing methods do not perform well when solving the system (4). Our main goal here is to find a fast and efficient preconditioner which can speed up the convergence of Krylov subspace methods by altering the spectral distribution of the original system in a desirable way [29, 30]. In [21], the following symmetric indefinite constraint preconditioner was constructed:

$$P_h = \begin{bmatrix} 0 & L_h^\intercal \\ L_h & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix}.$$

Notice that here $L_h^{-1}v$ (or $L_h^{-\intercal}v$) can be quickly computed by implementing a block forward (or backward) substitution, with the well-known Fast Fourier Transform (FFT) algorithm applied to each diagonal block. In particular, the preconditioning step $P_h^{-1}v$ can be computed with order $NM_1M_2\log(M_1M_2)$ operations in the 2D case. From the structural entries of L_h in P_h, we know that P_h is nonsingular. Moreover, the right preconditioned system is given by

$$A_h P_h^{-1} = \begin{bmatrix} I_h + \frac{1}{\gamma}\check{I}_h L_h^{-1}\hat{I}_h L_h^{-\intercal} & \check{I}_h L_h^{-1} \\ 0 & I_h \end{bmatrix},$$

where I_h corresponds to the identity matrix I applied at every time-step. Clearly, half of the eigenvalues of $A_h P_h^{-1}$ are exactly ones, while the remaining half are determined by

$$R_h := I_h + \frac{1}{\gamma}\check{I}_h L_h^{-1}\hat{I}_h L_h^{-\intercal}.$$

By exploring the connection between the matrices L_h^{-1}, $L_h^{-\intercal}$ and the underlying discretized linear system we have obtained the following theorem on the eigenvalue bounds of the preconditioned system:

Theorem 2.1. *[21, Theorem 3.4] Let $\lambda(R_h)$ be any eigenvalue of R_h, then $\lambda(R_h)$ is real and it satisfies*

$$1 < \lambda(R_h) < 1 + \frac{\kappa}{\gamma},$$

where κ is a positive constant that is independent of τ and h.

The above theorem states that all eigenvalues of the preconditioned coefficient matrix $A_h P_h^{-1}$ are uniformly greater than one, and are less than an upper bound that depends only on the regularization parameter γ and possibly the time period T. This predicts a mesh-independent convergence rate of the preconditioned Krylov subspace methods. Nevertheless, according to the above estimates, preconditioned Krylov methods may show a severely deteriorating convergence rate as the regularization parameter γ decreases towards zero, which is clearly observed in numerical simulations. To address this issue, we propose in the next section a new preconditioner that exhibits more robust convergence rates with respect to the possibly very small values of γ.

3. A New Preconditioner with Eigenvalue Analysis

We now derive and analyze a new preconditioner based on the finite difference approach described above. We highlight that such a preconditioner could easily be transferred to the finite element setting, with the identity matrix I_h replaced by a finite element mass matrix, and the Laplacian operator $-\Delta_h$ by a finite element stiffness matrix.

Inspired by the idea of matching Schur complement approximations, see e.g., [9, 11, 27, 31], we propose a new Schur-complement type preconditioner B_h in the following factorization form:

$$B_h = \begin{bmatrix} S_h - \gamma L_h^\intercal \hat{I}_h^{-1} L_h & L_h^\intercal \\ L_h & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix} = \begin{bmatrix} I_h & -\gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix} \begin{bmatrix} S_h & 0 \\ 0 & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix} \begin{bmatrix} I_h & 0 \\ -\gamma \hat{I}_h^{-1} L_h & I_h \end{bmatrix},$$

which is intended to well approximate the Schur complement decomposition of A_h

$$A_h = \begin{bmatrix} \check{I}_h & L_h^\intercal \\ L_h & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix} = \begin{bmatrix} I_h & -\gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix} \begin{bmatrix} \check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h & 0 \\ 0 & -\frac{1}{\gamma}\hat{I}_h \end{bmatrix} \begin{bmatrix} I_h & 0 \\ -\gamma \hat{I}_h^{-1} L_h & I_h \end{bmatrix},$$

in view of the following 'matching Schur complement' approximation[4]:

$$S_h := (\check{I}_h^{1/2}\hat{I}_h^{1/2} + \sqrt{\gamma}L_h^\intercal)\hat{I}_h^{-1}(\hat{I}_h^{1/2}\check{I}_h^{1/2} + \sqrt{\gamma}L_h) \approx \check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h. \tag{6}$$

This further implies that

$$S_h - \gamma L_h^\intercal \hat{I}_h^{-1} L_h \approx \check{I}_h \quad \text{or} \quad B_h \approx A_h.$$

Due to the facts that both $\hat{I}_h$ and $\check{I}_h$ are positive diagonal (close to an identity matrix), and L_h is block lower triangular, the operation costs of the preconditioning step $B_h^{-1}v$ are similar to, but slightly higher than, that of $P_h^{-1}v$ as explained above. Notice that the above preconditioner B_h is constructed based on the

[4]The term 'matching' refers to the fact that both terms of the exact Schur complement are captured within the approximation. In more detail, the multiplication of $(\check{I}_h^{1/2}\hat{I}_h^{1/2})\hat{I}_h^{-1}(\hat{I}_h^{1/2}\check{I}_h^{1/2})$ leads to the first term $\check{I}_h$ on the right of the expression (6), with the second term $\gamma L_h^\intercal \hat{I}_h^{-1} L_h$ obtained by the multiplication of $(\sqrt{\gamma}L_h^\intercal)\hat{I}_h^{-1}(\sqrt{\gamma}L_h)$.

Schur complement of the $(2,2)$ block of A_h. The construction of a similar preconditioner based on the Schur complement of the $(1,1)$ block of A_h is straightforward and hence for simplicity not discussed further.

Eigenvalues of the preconditioned system: Lower bounds

To estimate the eigenvalue bounds of the right preconditioned system $A_h B_h^{-1}$, we examine the matrix product

$$A_h B_h^{-1} = \begin{bmatrix} I_h & -\gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix} \begin{bmatrix} (\check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h) S_h^{-1} & 0 \\ 0 & I_h \end{bmatrix} \begin{bmatrix} I_h & -\gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix}^{-1}, \tag{7}$$

which implies that half of the eigenvalues of $A_h B_h^{-1}$ are ones, while the remaining half are given by the eigenvalues of the preconditioned Schur complement, i.e.,

$$Q_h := (\check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h) S_h^{-1}.$$

We highlight that as the preconditioner B_h (or P_h) is not symmetric positive definite, it is not possible to apply an iterative method such as MINRES [32] to solve the matrix system (4), and thus eigenvalue bounds for Q_h do not entirely describe the theoretical convergence rates of the non-symmetric method required. However, in practice, such estimates will provide a strong indication as to the effectiveness of our preconditioning approach, and will describe many of the convergence properties that we observe numerically.

To estimate bounds for the eigenvalues of Q_h, we highlight that this is similar to the case discussed in [31], but the conclusion reached does not directly apply to the matrices highlighted in this paper, due to the differing structures and algebraic properties of this matrix system. More specifically, in our case we have that $\check{I}_h \neq \hat{I}_h$, and that $L_h + L_h^\intercal$ is indefinite, which therefore will lead to eigenvalues spreading out of the desirable uniform interval $[\frac{1}{2}, 1]$ obtained in [31, Theorem 1]. For the upcoming analysis, we first recall the following lemma that provides a uniform lower bound for the eigenvalues of the preconditioned Schur complement for a very general matrix form.

Lemma 3.1. *[27, Theorem 1] Let W and $\widehat{W}$ be nonsingular matrices*

$$W = X^\intercal X + Y^\intercal Y, \quad \widehat{W} = (X+Y)^\intercal (X+Y),$$

with two given real matrices X and Y. Then all eigenvalues of $W\widehat{W}^{-1}$ are real and bounded below by $\frac{1}{2}$.

In the above preconditioned Schur complement Q_h, by further defining

$$X = \check{I}_h^{1/2}, \quad Y = \sqrt{\gamma} \hat{I}_h^{-1/2} L_h,$$

we arrive at the following relations:

$$\check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h = X^\intercal X + Y^\intercal Y =: W,$$

and

$$S_h = (\check{I}_h^{1/2} \hat{I}_h^{1/2} + \sqrt{\gamma} L_h^\intercal) \hat{I}_h^{-1} (\hat{I}_h^{1/2} \check{I}_h^{1/2} + \sqrt{\gamma} L_h) = (X+Y)^\intercal (X+Y) =: \widehat{W}.$$

Hence, we have obtained

$$Q_h = (\check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h) S_h^{-1} = W \widehat{W}^{-1},$$

which, by Lemma 3.1, implies any eigenvalue of Q_h, denoted by $\lambda(Q_h)$, satisfies $\lambda(Q_h) \geq \frac{1}{2}$.

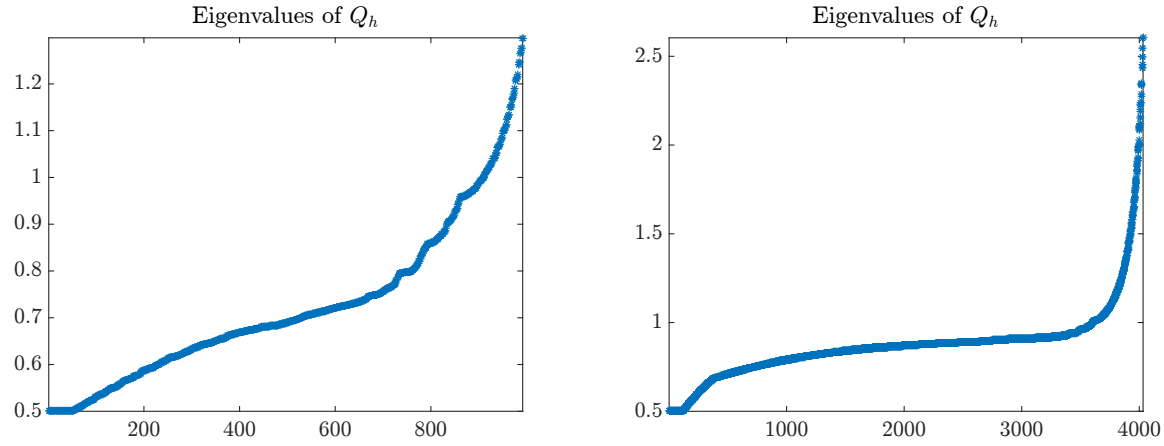

Figure 1: Eigenvalue distributions of Q_h in Example 1 with $M = N = 32$ (left) and $M = N = 64$ (right), respectively, and $\gamma = 10^{-6}$.

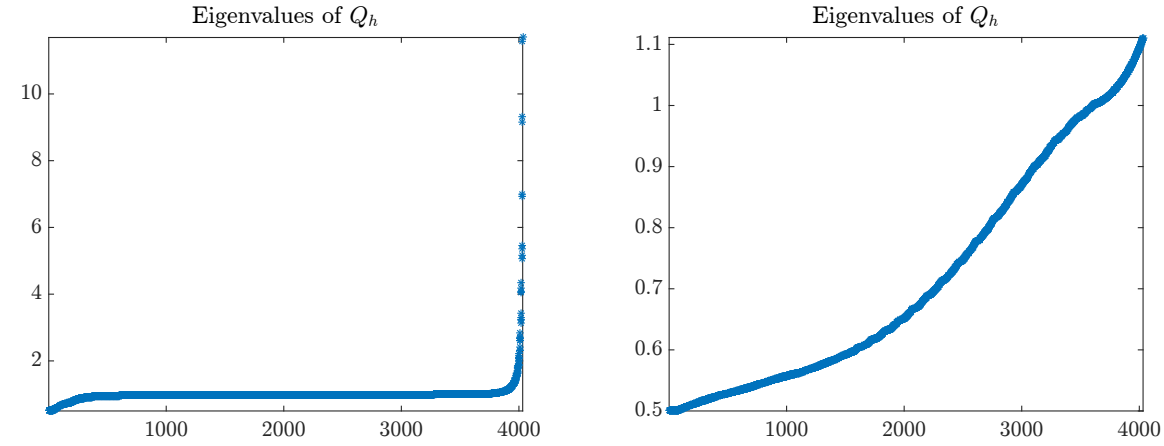

Figure 2: Eigenvalue distributions of Q_h in Example 1 with $\gamma = 10^{-4}$ (left) and $\gamma = 10^{-8}$ (right), respectively, and $M = N = 64$.

To provide a quick glimpse of the possible eigenvalue bounds for Q_h, it is often helpful to numerically visualize the distribution of the eigenvalues of Q_h. We observe in practice that the eigenvalues of Q_h are bounded within a slightly larger interval than $[\frac{1}{2}, 1]$, and the majority of them are actually located within this range. This hence predicts a robust convergence rate that is (nearly) independent with respect to the mesh step-size h and the regularization parameter γ. In Figure 1, we plot the numerically computed eigenvalues of Q_h for Example 1 (defined in Section 4) with $\gamma = 10^{-6}$ using $M = N = 32$ and $M = N = 64$, respectively. As expected, the eigenvalues of Q_h have a tight lower bound $\frac{1}{2}$, and a slightly increasing upper bound. Furthermore, in Figure 2, we plot the numerically computed eigenvalues of Q_h for Example 1 using $M = N = 64$ with $\gamma = 10^{-4}$ and $\gamma = 10^{-8}$, respectively. The distribution of eigenvalues becomes even more

clustered around $[0.5, 1.1]$ for $\gamma = 10^{-8}$, which hence indicates a faster convergence rate. In the following analysis, we will mathematically estimate these convenient eigenvalue bounds.

The symmetric part of L_h

To facilitate the upcoming analysis, we first derive a bound for estimating the eigenvalues of the symmetric part of L_h, $L_{\text{sym}} := L_h + L_h^\intercal$. Let $D_h = I - \frac{\tau^2}{2}\Delta_h =: I + \tau^2 K$, with $K = -\frac{1}{2}\Delta_h$ being a positive definite matrix. Explicitly, we have

$$
\begin{aligned}
L_{\text{sym}} &:= L_h + L_h^\intercal \qquad (8)\\
&= \frac{1}{\tau^2}\begin{bmatrix}
2D_h & -2I & D_h & 0 & 0 & \cdots & 0\\
-2I & 2D_h & -2I & D_h & 0 & \cdots & 0\\
D_h & -2I & 2D_h & -2I & D_h & \cdots & 0\\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\
0 & 0 & D_h & -2I & 2D_h & -2I & D_h\\
0 & 0 & \cdots & D_h & -2I & 2D_h & -2I\\
0 & 0 & 0 & \cdots & D_h & -2I & 2D_h
\end{bmatrix}\\
&= \frac{1}{\tau^2}\begin{bmatrix}
2I & -2I & I & 0 & 0 & \cdots & 0\\
-2I & 2I & -2I & I & 0 & \cdots & 0\\
I & -2I & 2I & -2I & I & \cdots & 0\\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\
0 & 0 & I & -2I & 2I & -2I & I\\
0 & 0 & \cdots & I & -2I & 2I & -2I\\
0 & 0 & 0 & \cdots & I & -2I & 2I
\end{bmatrix} \qquad (9)\\
&\quad + \begin{bmatrix}
2K & 0 & K & 0 & 0 & \cdots & 0\\
0 & 2K & 0 & K & 0 & \cdots & 0\\
K & 0 & 2K & 0 & K & \cdots & 0\\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\
0 & 0 & K & 0 & 2K & 0 & K\\
0 & 0 & \cdots & K & 0 & 2K & 0\\
0 & 0 & 0 & \cdots & K & 0 & 2K
\end{bmatrix}\\
&= \text{pendiag}([1,-2,2,-2,1]) \otimes \frac{1}{\tau^2} I + \text{pendiag}([1,0,2,0,1]) \otimes K\\
&=: C \otimes \frac{1}{\tau^2} I + F \otimes K,
\end{aligned}
$$

where pendiag(v) denotes a block Toeplitz $N \times N$ matrix with values of v on the five central block diagonals.

First of all, we point out that the eigenvalues of $K = -\frac{1}{2}\Delta_h$ are given as follows in the case $h_1 = h_2 = h$ (see e.g., [33, p.59]):

- **(1D case):** $h = \frac{1}{M}$; we focus on this case in the following analysis:

$$\lambda_j(K) = \frac{2}{h^2}\sin^2\left(\frac{j\pi h}{2}\right), \qquad j = 1, 2, ..., M-1.$$

- **(2D case):** $h_1 = h_2 = h = \frac{1}{M}$, with $M = M_1 = M_2$:

$$\lambda_{i,j}(K) = \frac{2}{h^2}\left(\sin^2\left(\frac{i\pi h}{2}\right) + \sin^2\left(\frac{j\pi h}{2}\right)\right), \qquad i,j = 1,2,...,M-1.$$

A simple calculation can verify that

$$C := \text{pendiag}([1,-2,2,-2,1]) = \Psi^2 - 2\Psi + e_1 e_1^\intercal + e_N e_N^\intercal,$$

where $\Psi = \text{tridiag}([-1,2,-1]) \in \mathbb{R}^{N\times N}$, and $e_i \in \mathbb{R}^N$ is the i-th column of an identity matrix. Recall the well-known formula of the eigenvalues of the tridiagonal Toeplitz matrix Ψ (see [34, 35, 36, 37], for example):

$$\lambda_k(\Psi) = 2 - 2\cos(k\pi l) = 4\sin^2\left(\frac{k\pi l}{2}\right), \qquad k = 1,2,...,N,$$

where $l = \frac{1}{N+1}$ is based on the matrix size, and is in general different from the time-step size $\tau = \frac{T}{N}$. This then leads to the observation

$$\lambda_k(C) \geq \lambda_k(\Psi^2 - 2\Psi) = \lambda_k^2(\Psi) - 2\lambda_k(\Psi) = (\lambda_k(\Psi) - 1)^2 - 1 = (1 - 2\cos(k\pi l))^2 - 1 \geq -1,$$

since $e_1 e_1^\intercal + e_N e_N^\intercal$ is positive semidefinite, with $\lambda_{\min}(e_1 e_1^\intercal + e_N e_N^\intercal) = 0$. Another similar calculation gives that

$$F := \text{pendiag}([1,0,2,0,1]) = E^2 + e_1 e_1^\intercal + e_N e_N^\intercal,$$

where $E = \text{tridiag}([1,0,1]) \in \mathbb{R}^{N\times N}$. Again, by the well-known formula for tridiagonal Toeplitz matrices (see [36, 37]), we have

$$\lambda_k(E) = -2\cos(k\pi l), \qquad k = 1,2,...,N,$$

which then leads to

$$\lambda_k(F) \geq \lambda_k(E^2) = 4\cos^2(k\pi l) > 0,$$

since $\lambda_{\min}(e_1 e_1^\intercal + e_N e_N^\intercal) = 0$. Putting together all of the lower bounds derived above, we obtain that

$$\lambda_{k,j}(L_{\text{sym}}) = \frac{1}{\tau^2}\lambda_k(C) + \lambda_k(F)\lambda_j(K) > -\frac{1}{\tau^2}.$$

In view of the eigenvalue expressions of C, F, K, the above lower bound is very tight. For convenience, we summarize the above arguments in the following interesting result:

Theorem 3.1. *Let $L_{\text{sym}} := L_h + L_h^\intercal$ be the symmetric part of L_h defined in (5). Then it holds that*

$$\lambda(L_{\text{sym}}) > -\frac{1}{\tau^2},$$

where the lower bound is independent of h.

Estimating an upper bound for the eigenvalues of the preconditioned system

Deriving an upper bound for the eigenvalues of Q_h turns out to be more challenging than the lower bound, mainly due to the fact that the matrix

$$X^\intercal Y + Y^\intercal X = \sqrt{\gamma}\left(\check{I}_h^{1/2}\hat{I}_h^{-1/2}L_h + (\check{I}_h^{1/2}\hat{I}_h^{-1/2}L_h)^\intercal\right)$$

is indefinite for the matrices under consideration. To estimate an upper bound for eigenvalues of $Q_h = W\widehat{W}^{-1}$, we consider the range of the corresponding Rayleigh quotient. For any real vector $v \neq 0$, we define $a = Xv$ and $b = Yv$, and write:

$$\mu := \frac{v^\intercal W v}{v^\intercal \widehat{W} v} = \frac{a^\intercal a + b^\intercal b}{a^\intercal a + b^\intercal b + a^\intercal b + b^\intercal a},$$

which gives bounds for the eigenvalues of $Q_h = W\widehat{W}^{-1}$. Clearly, if $X^\intercal Y + Y^\intercal X$ is positive semidefinite, then $a^\intercal b + b^\intercal a = v^\intercal(X^\intercal Y + Y^\intercal X)v \geq 0$ and hence $\mu \leq 1$. For more complex problems such as the PDE system under consideration, where this property does not hold, we may rewrite

$$\mu = \left(1 + \frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b}\right)^{-1},$$

which will lead to an upper bound

$$\mu \leq \frac{1}{1+\omega}$$

if we can find a lower bound of the form

$$0 > \frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b} \geq \omega > -1,$$

where one only needs to bound values such that $a^\intercal b + b^\intercal a < 0$.

We start by estimating a lower bound for the simplified Rayleigh quotient (dropping $b^\intercal b$):

$$\frac{a^\intercal b + b^\intercal a}{a^\intercal a} = \frac{v^\intercal(X^\intercal Y + Y^\intercal X)v}{v^\intercal(X^\intercal X)v} = \sqrt{\gamma}\frac{v^\intercal\left(\check{I}_h^{1/2}\hat{I}_h^{-1/2}L_h + (\check{I}_h^{1/2}\hat{I}_h^{-1/2}L_h)^\intercal\right)v}{v^\intercal(\check{I}_h^{1/2}\check{I}_h^{1/2})v},$$

which is equivalent to finding a lower bound for the smallest eigenvalue of the following matrix:

$$Q_a := \sqrt{\gamma}\check{I}_h^{-1/2}\left(\check{I}_h^{1/2}\hat{I}_h^{-1/2}L_h + L_h^\intercal\hat{I}_h^{-1/2}\check{I}_h^{1/2}\right)\check{I}_h^{-1/2}.$$

Using the definitions of the matrices involved, we easily obtain that

$$
\begin{aligned}
Q_a &= \sqrt{\gamma}\left(\hat{I}_h^{-1/2} L_h \check{I}_h^{-1/2} + \check{I}_h^{-1/2} L_h^\intercal \hat{I}_h^{-1/2}\right) \\
&= \frac{\sqrt{\gamma}}{\tau^2}\begin{bmatrix}
2\sqrt{2}D_h & -2I & D_h & 0 & 0 & \cdots & 0 \\
-2I & 2D_h & -2I & D_h & 0 & \cdots & 0 \\
D_h & -2I & 2D_h & -2I & D_h & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & D_h & -2I & 2D_h & -2I & D_h \\
0 & 0 & \cdots & D_h & -2I & 2D_h & -2I \\
0 & 0 & 0 & \cdots & D_h & -2I & 2\sqrt{2}D_h
\end{bmatrix} \\
&= \frac{\sqrt{\gamma}}{\tau^2}\begin{bmatrix}
2D_h & -2I & D_h & 0 & 0 & \cdots & 0 \\
-2I & 2D_h & -2I & D_h & 0 & \cdots & 0 \\
D_h & -2I & 2D_h & -2I & D_h & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & D_h & -2I & 2D_h & -2I & D_h \\
0 & 0 & \cdots & D_h & -2I & 2D_h & -2I \\
0 & 0 & 0 & \cdots & D_h & -2I & 2D_h
\end{bmatrix} \\
&\quad + \frac{2(\sqrt{2}-1)\sqrt{\gamma}}{\tau^2}\begin{bmatrix}
D_h & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & D_h
\end{bmatrix} \qquad (10) \\
&=: \sqrt{\gamma}\, L_{\text{sym}} + G, \qquad (11)
\end{aligned}
$$

which leads to the following lower bound (using Theorem 3.1, and noticing that the right matrix G in (11) is positive semi-definite):

$$
\lambda(Q_a) \geq \sqrt{\gamma}\, \lambda_{\min}(L_{\text{sym}}) + \lambda_{\min}(G) \geq \sqrt{\gamma}\, \lambda_{\min}(L_{\text{sym}}) > -\frac{\sqrt{\gamma}}{\tau^2}.
$$

In terms of the Rayleigh quotient, we have in fact obtained the following inequalities (for negative values of $a^\intercal b + b^\intercal a$):

$$
\frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b} \geq \frac{a^\intercal b + b^\intercal a}{a^\intercal a} > -\frac{\sqrt{\gamma}}{\tau^2}. \qquad (12)
$$

On the other hand, since $a = Xv \neq 0$, $b = Yv \neq 0$, $a + b = (X+Y)v \neq 0$, we always have that

$$
0 < (a+b)^\intercal (a+b) = a^\intercal a + b^\intercal b + a^\intercal b + b^\intercal a \qquad \Longleftrightarrow \qquad \frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b} > -1.
$$

Therefore, we obtain the bound

$$
\frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b} > -\min\left\{1, \frac{\sqrt{\gamma}}{\tau^2}\right\} =: \omega_1(\gamma, \tau).
$$

Based on this, we can establish the eigenvalue bounds for Q_h:

$$\frac{1}{2} \leq \mu < \frac{1}{1+\omega_1(\gamma,\tau)}.$$

This Rayleigh quotient (or eigenvalue) bound tells us that:

- the convergence rate of the Krylov subspace solver should be independent of the spatial mesh step-size h;
- for a fixed τ, we should observe a faster convergence rate as γ tends to zero (less than $O(\tau^4)$);
- when $\gamma \geq \tau^4$ (with either large γ or small τ), this does not provide a uniform upper bound.

A refined upper bound for the eigenvalues

Numerically, with a fixed τ, we observe significantly faster convergence rates as γ becomes very large, which is not explained or predicted by the above eigenvalue bounds. To understand such improved convergence rates, we further improve the above estimate by making use of the result of Theorem 2.1:

$$1 < \lambda\left(I_h + \frac{1}{\gamma}\check{I}_h L_h^{-1}\hat{I}_h L_h^{-\intercal}\right) = \lambda\left((\gamma L_h^{\intercal}\hat{I}_h^{-1}L_h + \check{I}_h)(\gamma L_h^{\intercal}\hat{I}_h^{-1}L_h)^{-1}\right) < 1 + \frac{\kappa}{\gamma},$$

where $\kappa > 0$ is a constant independent of τ and h. In terms of the above notation, this corresponds exactly to the observation

$$1 < \lambda\left((Y^{\intercal}Y + X^{\intercal}X)(Y^{\intercal}Y)^{-1}\right) < 1 + \frac{\kappa}{\gamma},$$

which is equivalent to the following Rayleigh quotient bounds:

$$1 < \frac{a^{\intercal}a + b^{\intercal}b}{b^{\intercal}b} < 1 + \frac{\kappa}{\gamma}.$$

From this we can easily obtain that

$$\frac{1}{1+\frac{\kappa}{\gamma}} < \frac{b^{\intercal}b}{a^{\intercal}a + b^{\intercal}b} < 1,$$

and hence

$$0 < \frac{a^{\intercal}a}{a^{\intercal}a + b^{\intercal}b} = 1 - \frac{b^{\intercal}b}{a^{\intercal}a + b^{\intercal}b} < 1 - \frac{1}{1+\frac{\kappa}{\gamma}} = \frac{1}{1+\frac{\gamma}{\kappa}}.$$

Therefore, we further see that

$$0 > \frac{a^{\intercal}b + b^{\intercal}a}{a^{\intercal}a + b^{\intercal}b} = \frac{a^{\intercal}b + b^{\intercal}a}{a^{\intercal}a} \cdot \frac{a^{\intercal}a}{a^{\intercal}a + b^{\intercal}b} > -\frac{\sqrt{\gamma}}{\tau^2} \cdot \frac{1}{1+\frac{\gamma}{\kappa}},$$

using (12), which finally gives an improved bound

$$\frac{1}{2} \leq \mu < \frac{1}{1+\omega_2(\gamma,\tau)} \leq \frac{1}{1+\omega_1(\gamma,\tau)},$$

with

$$\omega_2(\gamma,\tau) := -\min\left\{1, \frac{\sqrt{\gamma}}{\tau^2} \cdot \frac{1}{1+\frac{\gamma}{\kappa}}\right\} \geq \omega_1(\gamma,\tau).$$

This improved bound implies that, for a fixed $\tau > 0$, we should obtain a faster convergence rate as γ becomes significantly large or small. A rough estimate indicates that we should observe particularly fast convergence whenever $\gamma < \tau^4$ or $\gamma > \frac{\kappa^2}{\tau^4}$. However, the convergence rates with $\gamma \in (\tau^4, \frac{\kappa^2}{\tau^4})$ are not fully explained by the above eigenvalue estimates. We point out that the relation $\gamma \sim \tau^4$ also arises in a technical assumption for the proof of the original error estimate of the implicit central finite difference scheme used in [21], although we note that the numerical results in [21] suggest that the scheme often achieves second-order accuracy even without such a condition ($\gamma \geq \tau^4$). Based on our numerical experiments, however, the required number of iterations in fact increases very mildly in practice when $\gamma \geq \tau^4$.

To summarize, as one of our main theoretical conclusions, we have obtained the following key estimate:

Theorem 3.2. *Let $\lambda(A_h B_h^{-1})$ be any eigenvalue of $A_h B_h^{-1}$, then it holds that*

$$\frac{1}{2} \leq \lambda(A_h B_h^{-1}) < \frac{1}{1 + \omega_2(\gamma, \tau)},$$

with

$$\omega_2(\gamma, \tau) = -\min\left\{1, \frac{\sqrt{\gamma}}{\tau^2} \cdot \frac{1}{1 + \frac{\gamma}{\kappa}}\right\}.$$

In particular, this eigenvalue bound is independent of h.

A further estimate as $\tau \to 0$

To obtain more comprehensive eigenvalue estimates for the challenging case $\gamma \geq \tau^4$, we wish to bound the following full Rayleigh quotient:

$$\frac{a^\intercal b + b^\intercal a}{a^\intercal a + b^\intercal b} = \frac{v^\intercal (X^\intercal Y + Y^\intercal X) v}{v^\intercal (X^\intercal X + Y^\intercal Y) v} = \sqrt{\gamma} \frac{v^\intercal \left(\check{I}_h^{1/2} \hat{I}_h^{-1/2} L_h + (\check{I}_h^{1/2} \hat{I}_h^{-1/2} L_h)^\intercal \right) v}{v^\intercal (\check{I}_h + \gamma L_h^\intercal \hat{I}_h^{-1} L_h) v},$$

which is very intricate to estimate directly. Upon applying the approximations of replacing $\check{I}_h$ and $\hat{I}_h$ by I_h, we obtain the simplified Rayleigh quotient

$$r_s := \sqrt{\gamma} \frac{v^\intercal \left(L_h + L_h^\intercal \right) v}{v^\intercal (I_h + \gamma L_h^\intercal L_h) v}.$$

Notice that we have a Kronecker product formulation for L_h, i.e.,

$$L_h = \frac{1}{\tau^2} \text{pendiag}([1, -2, 1, 0, 0]) \otimes I + \text{pendiag}([1, 0, 1, 0, 0]) \otimes K =: \frac{1}{\tau^2} Z_N \otimes I + V_N \otimes K.$$

With this simpler formulation, we can easily observe that

$$\begin{aligned} L_h + L_h^\intercal &= \frac{1}{\tau^2} \text{pendiag}([1, -2, 2, -2, 1]) \otimes I + \text{pendiag}([1, 0, 2, 0, 1]) \otimes K \\ &= \frac{1}{\tau^2} (Z_N + Z_N^\intercal) \otimes I + (V_N + V_N^\intercal) \otimes K \end{aligned}$$

and

$$L_h^\intercal L_h = \left(\frac{1}{\tau^2} Z_N^\intercal \otimes I + V_N^\intercal \otimes K\right)\left(\frac{1}{\tau^2} Z_N \otimes I + V_N \otimes K\right)$$
$$= \frac{1}{\tau^4}(Z_N^\intercal Z_N) \otimes I + \frac{1}{\tau^2}(Z_N^\intercal V_N + V_N^\intercal Z_N) \otimes K + (V_N^\intercal V_N) \otimes K^2.$$

We are particularly interested in the challenging situation of $\tau \to 0$ with fixed h and γ. To gain an idea of what might occur in this situation, we consider dropping these small terms in τ (relative to $O(\frac{1}{\tau^2})$) to reach the approximations

$$L_h + L_h^\intercal \approx \frac{1}{\tau^2}(Z_N + Z_N^\intercal) \otimes I, \qquad I_h + \gamma L_h^\intercal L_h \approx \frac{\gamma}{\tau^4}(Z_N^\intercal Z_N) \otimes I,$$

which then leads to the following approximation for the matrix arising from r_s:

$$\sqrt{\gamma}\,(L_h + L_h^\intercal)\,(I_h + \gamma L_h^\intercal L_h)^{-1} \approx \frac{\tau^2}{\sqrt{\gamma}}\left((Z_N + Z_N^\intercal)(Z_N^\intercal Z_N)^{-1}\right) \otimes I.$$

Numerically, we observe a very clustered distribution for the eigenvalues of the key matrix

$$\tau^2(Z_N + Z_N^\intercal)(Z_N^\intercal Z_N)^{-1} \simeq \tau^2 Z_N^{-\intercal}(Z_N + Z_N^\intercal)Z_N^{-1} = \tau^2(Z_N^{-\intercal} + Z_N^{-1}) =: \mathcal{Z},$$

which hence indicates a good (but possibly τ-dependent) convergence rate as $\tau \to 0$. Here $\simeq$ denotes similarity between two matrices, and we have used the fact that $\bar{A}\bar{B} \simeq \bar{B}\bar{A}$ for any invertible matrices $\bar{A}$ and $\bar{B}$.

Recall that $Z_N = \text{pendiag}([1, -2, 1, 0, 0])$ has a dimension $N \times N$ with $\tau = \frac{T}{N}$, therefore giving

$$\mathcal{Z} = \frac{T^2}{N^2}(Z_N^{-\intercal} + Z_N^{-1}).$$

Simple mathematical induction (see Lemma A.1 in Appendix A) shows that Z_N^{-1} is a dense lower triangular Toeplitz matrix of the form

$$Z_N^{-1} = \text{LTToeplitz}([1, 2, \cdots, N-1, N]^\intercal),$$

in which $[1, 2, \cdots, N-1, N]^\intercal$ gives the first column of Z_N^{-1}. Notice that Z_N is a sparse lower triangular Toeplitz matrix. Hence we can exactly compute the infinity norm of $\mathcal{Z}$

$$\|\mathcal{Z}\|_\infty = \frac{T^2}{N^2}\|Z_N^{-\intercal} + Z_N^{-1}\|_\infty = \frac{T^2}{N^2}\left(\frac{N(N+1)}{2} + 1\right) = \frac{1}{2}\left(T^2 + T\tau + 2\tau^2\right),$$

which implies that (provided $\tau \le \frac{T}{2}$)

$$|\lambda(\mathcal{Z})| \le \|\mathcal{Z}\|_\infty = \frac{1}{2}\left(T^2 + T\tau + 2\tau^2\right) \le T^2.$$

Though with simplified arguments, this estimate does indeed provide asymptotic information for the Rayleigh quotient under consideration:

$$r_s \approx \frac{1}{\sqrt{\gamma}}|\lambda(\mathcal{Z})| \le \frac{T^2}{\sqrt{\gamma}}.$$

This partially explains the very good (though not mesh-independent in time) convergence rates observed as

$\tau \to 0$ (see also Table 4 below). At this point, our derived eigenvalue bounds are not completely independent of τ and γ, but the above analysis gives us useful guidance as to the effectiveness of our preconditioner, and indeed may lead to an improved preconditioner that can fulfill the goal of parameter independence. Based on our current understanding, it seems that the independence on τ and γ cannot be achieved simultaneously: we highlight once more that the constraint preconditioner P_h does not yield γ-independent convergence rates.

Some notes on GMRES convergence and an alternative preconditioner

As previously mentioned, due to the fact that our preconditioner B_h is not symmetric positive definite, we are unable to directly apply the preconditioned MINRES algorithm to solve the matrix system (4) involving A_h. Therefore, we elect to use the (right) preconditioned GMRES algorithm [38] with our preconditioner B_h. However, unlike Krylov methods such as MINRES (with a symmetric positive definite preconditioner), the eigenvalues of the preconditioned system alone do not conclusively determine the convergence rate of GMRES. To illustrate this we first note that, at the kth iteration of the GMRES method applied to a general matrix system $Ax = b$, the residual $r_k := b - Ax_k$ (with x_k denoting the kth iterate) satisfies [38]:

$$\|r_k\|_2 = \min_{q \in \Pi_k,\, q(0)=1} \|q(A) r_0\|_2 \,,$$

where Π_k is the set of polynomials of degree at most k. Supposing A is diagonalizable (i.e., $A = Z\Lambda Z^{-1}$ with a diagonal matrix Λ containing all the eigenvalues of A), it then follows that

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa_2(Z) \min_{q \in \Pi_k,\, q(0)=1} \max_{\lambda \in \Gamma(A)} |q(\lambda)|, \tag{13}$$

where $\kappa_2(Z) := \|Z\|_2 \|Z^{-1}\|_2$ denotes the condition number of Z and $\Gamma(A)$ denotes the spectrum of A. For the right preconditioned system matrix $A_h B_h^{-1}$ in (7), such a diagonalization will involve a block upper triangular matrix $V := \begin{bmatrix} I_h & -\gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix}$, due to this matrix appearing within the Schur complement decompositions of A_h and B_h. The condition number of V is then given by

$$\kappa_2(V) = \sqrt{\frac{\lambda_{\max}(V^\intercal V)}{\lambda_{\min}(V^\intercal V)}}.$$

Following the arguments in the proof of [39, Lemma 3.1], we find that the eigenvalues of $V^\intercal V$ are given by $1 + \frac{1}{2}\sigma_i(\sigma_i \pm \sqrt{\sigma_i^2 + 4})$, with σ_i the singular values of the off-diagonal matrix $(-\gamma L_h^\intercal \hat{I}_h^{-1})$. As this matrix becomes highly ill-conditioned when the mesh in space or time variables is refined, due in turn to the ill-conditioning of L_h, a typical convergence analysis for GMRES is unlikely to yield descriptive estimates when using this approach, and would not improve on the estimates we have derived using our eigenvalue analysis above. More specifically, noting that $V^{-1} = \begin{bmatrix} I_h & \gamma L_h^\intercal \hat{I}_h^{-1} \\ 0 & I_h \end{bmatrix}$ and using [39, Lemma 3.1], there holds

$$\kappa_2(V) = \|V\|_2 \|V^{-1}\|_2 = \phi(\| - \gamma L_h^\intercal \hat{I}_h^{-1}\|_2)\phi(\|\gamma L_h^\intercal \hat{I}_h^{-1}\|_2) = \phi^2(\|\gamma L_h^\intercal \hat{I}_h^{-1}\|_2),$$

where $\phi(t) = \sqrt{1 + \frac{1}{2}t(t + \sqrt{t^2+4})}$. Hence $\kappa_2(V) = O(\|L_h\|_2^2)$, with $\|L_h\|_2 \geq \frac{1}{\tau^2}\|D_h\|_2 = O(\tau^{-2}) + O(h^{-2})$ – the inequality arises from the fact that $\frac{1}{\tau^2} D_h = \frac{1}{\tau^2} I_h - \frac{1}{2}\Delta_h$ is a sub-matrix of L_h, combined with the analysis of [40, p. 143], whereupon the eigenvalues of Δ_h lead to the expression for the 2-norm.

It is worth noting, however, that although descriptive estimates of convergence are unlikely to be achieved using the bound (13), it is possible that in practice the theoretical convergence rate is not substantially worsened by the presence of the term $\kappa_2(Z)$. For example, if a bound of the form

$$\eta_k := \min_{q\in\Pi_k,\, q(0)=1} \max_{\lambda\in\Gamma(A)} |q(\lambda)| \leq \texttt{constant} \cdot \bar{q}^k$$

were stated for some fixed $\bar{q} \in (0,1)$, then the required number of GMRES iterations to achieve a specified level of residual reduction would depend at most logarithmically on $\kappa_2(Z)$. Although our analysis indicates that the value of η_k is itself influenced by the numerical discretization, this suggests that the distribution of eigenvalues of $A_h B_h^{-1}$ contributes substantially to the practical GMRES convergence rate. It remains to be further discussed regarding whether such a convenient upper bound of η_k can be established in general.

In the numerical results of the next section, we choose to apply the symmetric indefinite preconditioner B_h to solve the matrix system (4), as in practice we experience rapid and robust convergence for a range of problems, and we note that applying preconditioners which involve the constraint block L_h (and $L_h^\intercal$) of A_h frequently achieve more rapid convergence than preconditioners that do not include these terms. The drawback of this approach is the less rigorous convergence analysis one may carry out for the GMRES method, as discussed. Alternatively, it would also be possible to apply the following symmetric positive definite, block diagonal preconditioner:

$$\Theta_h = \begin{bmatrix} \check{I}_h & 0 \\ 0 & \frac{1}{\gamma} S_h \end{bmatrix}$$

within the MINRES algorithm, whereupon one may prove the convergence rate solely based on the eigenvalues of the preconditioned Schur complement, i.e., the eigenvalues of Q_h, thus circumventing these theoretical difficulties. We refer to [28, 41, 42] for further discussion of block diagonal preconditioners in general, and [9, 10, 11, 15] for their application to optimal control problems.

4. Numerical Examples

In this section, we provide several numerical examples to validate the theoretical analysis carried out, and to demonstrate the efficiency of our proposed preconditioner B_h. All simulations are implemented using MATLAB 2017b on a Dell Precision Workstation with Intel(R) Core(TM) i7-7700K CPU@4.2GHz and 32GB RAM. The CPU time (in seconds) is estimated using the timing functions `tic/toc`. We employ a standard right-preconditioned GMRES solver (without restarts) provided by the IFISS package [43, 44, 45], and choose a zero initial guess and a stopping tolerance `tol` based on the reduction in relative residual norms.

Example 1 [21]. Let $\Omega = (0,1)$ and $T = 2$. We choose $y_0(x) = \sin(\pi x)$, $y_1(x) = 0$,

$$f = -\frac{1}{\gamma}\sin(\pi x)(t-T)^2, \quad \text{and} \quad g(x,t) = 2\sin(\pi x) + \pi^2 \sin(\pi x)(t-T)^2 + \sin(\pi x)\cos(\pi t),$$

such that the exact solution of (1)–(2) is

$$y(x,t) = \sin(\pi x)\cos(\pi t) \quad \text{and} \quad p(x,t) = \sin(\pi x)(t-T)^2.$$

In Table 1, we report the required number of iterations and CPU times (in seconds) using the constraint preconditioner P_h for a wide range of regularization parameters γ. Here the iteration numbers for a fixed

mesh (corresponding to each row) increase significantly as γ tends to zero, although we do observe clear mesh-independent convergence for fixed γ (with a sufficiently fine mesh). Notice that the iteration numbers in the last two columns (i.e., the cases $\gamma = 10^{-10}, 10^{-12}$) should continue to increase if we apply a smaller tolerance `tol`, hence the obtained numerical solutions may not be accurate due to earlier termination of the iterative solvers. In those cases, the linear system has a very large condition number, which means that we should not expect to achieve very accurate numerical solutions, unless we choose the tolerance `tol` to be close to machine precision. This would lead to higher iteration numbers and computation times.

In Table 2, we report the required number of iterations and CPU times for the matching Schur complement preconditioner B_h. In comparison with the constraint preconditioner P_h, our new preconditioner B_h yields much faster convergence rates when the regularization parameter γ becomes smaller. Based on our eigenvalue analysis, we would expect to achieve faster convergence rates (or require fewer iterations) when the regularization parameter γ becomes very small, which is confirmed by the results in the final few columns of Table 2.

Table 1: Numbers of GMRES iterations and CPU times for Example 1 with the constraint preconditioner P_h.

`tol` $= 10^{-10}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(128,128)	8	0.04	18	0.08	61	0.34	137	1.02	144	1.10
(256,256)	8	0.14	18	0.33	62	1.77	151	8.31	152	8.68
(512,512)	8	0.52	18	1.32	63	7.49	159	35.55	154	33.73
(1024,1024)	8	1.86	18	4.83	64	32.70	162	156.94	155	142.16
(2048,2048)	8	7.57	18	20.07	65	136.26	163	704.66	157	644.36

Table 2: Numbers of GMRES iterations and CPU times for Example 1 with the matching Schur complement preconditioner B_h.

`tol` $= 10^{-10}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(128,128)	13	0.09	19	0.14	15	0.10	3	0.02	1	0.01
(256,256)	12	0.28	23	0.59	21	0.56	4	0.09	1	0.04
(512,512)	12	1.05	26	2.58	33	3.56	10	0.84	1	0.13
(1024,1024)	12	3.73	26	9.98	48	24.34	20	6.88	1	0.45
(2048,2048)	12	15.37	27	42.59	58	129.16	31	51.33	1	1.78

To further validate our eigenvalue analysis, we report more results from applying preconditioner B_h in different contexts:

- Table 3 shows the clear h-independent convergence by decreasing h, while fixing τ and γ.
- Table 4 shows the mildly τ-dependent convergence by decreasing τ, while fixing h and γ.
- Table 5 shows the number of iterations for both preconditioners by varying γ, while fixing h and τ.

Notice that the operation cost of one iteration of the matching Schur complement preconditioner B_h is slightly higher than that of the constraint preconditioner P_h, hence it is more appropriate to compare the CPU times in addition to the iteration numbers. Clearly, the convergence rate of P_h becomes much slower whenever we decrease γ, while the convergence rate of B_h exhibits the behavior predicted by our analysis. It is also interesting to observe that P_h yields a slightly faster convergence rate than B_h (though not dramatically so) when γ is greater than around 10^{-4}. Such a switching point, and the difference in convergence rates, can be

easily read from Figure 3, where the required number of iterations is treated as a function of the decreasing regularization parameter γ. The new preconditioner B_h yields much faster convergence rates for the cases with $\gamma \leq 10^{-5}$, while still providing comparable convergence rates for larger γ.

Table 3: Numbers of GMRES iterations and CPU times for Example 1 (fixing τ) with the matching Schur complement preconditioner B_h.

$\texttt{tol} = 10^{-10}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(128,1024)	12	0.69	26	1.64	48	3.47	20	1.16	1	0.10
(256,1024)	12	1.17	26	2.90	48	6.38	20	2.06	1	0.15
(512,1024)	12	2.07	26	5.34	47	11.76	20	3.85	1	0.26
(1024,1024)	12	3.77	26	9.93	48	24.10	20	7.04	1	0.46
(2048,1024)	12	7.56	26	20.05	48	48.15	20	14.05	1	0.90

Table 4: Numbers of GMRES iterations and CPU times for Example 1 (fixing h) with the matching Schur complement preconditioner B_h.

$\texttt{tol} = 10^{-10}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(1024,128)	13	0.53	20	0.98	15	0.59	2	0.10	1	0.07
(1024,256)	12	0.93	24	2.21	21	1.87	4	0.32	1	0.12
(1024,512)	12	1.88	26	4.96	34	7.02	10	1.53	1	0.24
(1024,1024)	12	3.75	26	9.97	48	24.23	20	6.90	1	0.46
(1024,2048)	12	7.57	27	21.52	58	66.01	30	24.65	1	0.92

Table 5: Numbers of GMRES iterations and CPU times for Example 1 (fixing $M = N = 128$) with a sequence of decreasing regularization parameter γ.

$\texttt{tol} = 10^{-14}$	Preconditioner P_h		Preconditioner B_h	
γ	Iter	CPU	Iter	CPU
10^{1}	5	0.03	9	0.07
10^{0}	6	0.03	10	0.08
10^{-1}	8	0.04	13	0.08
10^{-2}	10	0.04	16	0.11
10^{-3}	15	0.07	23	0.16
10^{-4}	28	0.13	30	0.20
10^{-5}	53	0.27	32	0.29
10^{-6}	99	0.63	25	0.16
10^{-7}	177	1.52	16	0.10
10^{-8}	287	3.30	10	0.07
10^{-9}	394	5.64	6	0.04
10^{-10}	461	7.41	3	0.02
10^{-11}	499	8.49	2	0.02
10^{-12}	532	9.61	1	0.01

Example 2 [21]. Let $\Omega = (0,1)^2$ and $T = 2$. We choose

$$y_0(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2), \quad y_1(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2),$$

$$f(x_1, x_2, t) = (1 + 2\pi^2)e^t \sin(\pi x_1)\sin(\pi x_2) - \frac{1}{\gamma}(t - T)^2 \sin(\pi x_1)\sin(\pi x_2),$$

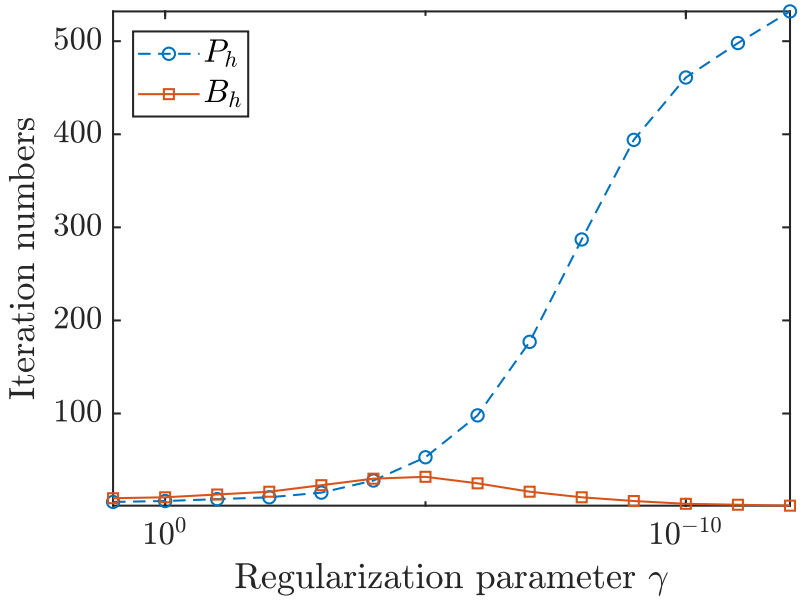

Figure 3: Comparison of iteration numbers with preconditioners P_h and B_h in Example 1 (with fixed $M = N = 128$).

and

$$g(x_1, x_2, t) = (e^t + 2 + 2\pi^2(t-T)^2)\sin(\pi x_1)\sin(\pi x_2),$$

such that the exact solution of (1)–(2) is

$$y(x_1, x_2, t) = e^t \sin(\pi x_1)\sin(\pi x_2) \quad \text{and} \quad p(x_1, x_2, t) = (t-T)^2 \sin(\pi x_1)\sin(\pi x_2).$$

For this example, we deliberately use a slightly larger tolerance `tol` (typically 10^{-8}) and a maximum of 100 iterations in the GMRES solver, to control the overall computation time. It is worthwhile to point out that a mesh size of (256,256,256) in fact leads to roughly 34 million unknowns, which is highly nontrivial to solve using any black-box iterative or sparse direct solver.

In Tables 6 and 7, we report the required number of iterations and CPU times for solving Example 2 using the constraint preconditioner P_h and the matching Schur complement preconditioner B_h, respectively. Similarly, Tables 8 and 9 show the expected h-independent convergence and the mildly τ-dependent convergence of the preconditioner B_h, respectively. Finally, Table 10 shows the number of iterations for both preconditioners by increasing/decreasing γ, while fixing h and τ, which are also plotted in Figure 4 for better visualization. Based on these tables, we observe very similar convergence rates as seen in Example 1, faithfully corresponding to our theoretical analysis which is also valid for both the 2D and 3D cases. In particular, the new proposed matching Schur complement preconditioner B_h converges significantly faster than the constraint preconditioner P_h when $\gamma \leq 10^{-5}$.

Example 3 [21]. To extend the reach of our solvers to more practical wave control problems, in this example we numerically investigate the applicability and efficiency of our proposed preconditioner in the case where additional constraints are applied to the control (see, e.g., [46] and references therein). More specifically, we test our methods on problems involving the box constraints $u \in U_{ad} := \{u \in U \mid u_a \leq u \leq u_b\}$, or the one-sided constraint $u \in U_{ad} := \{u \in U \mid u_a \leq u\}$. We treat the resulting non-smooth optimality

Table 6: Numbers of GMRES iterations and CPU times for Example 2 with the constraint preconditioner P_h.

$\texttt{tol} = 10^{-8}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M_1, M_2, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(16,16,16)	8	0.03	11	0.07	17	0.09	17	0.07	17	0.08
(32,32,32)	8	0.09	12	0.15	35	0.39	37	0.57	36	0.45
(64,64,64)	8	0.70	12	1.07	44	5.32	47	5.95	47	6.47
(128,128,128)	7	5.01	12	9.14	46	56.26	49	51.28	49	51.51
(256,256,256)	7	52.08	13	100.25	48	513.58	50	542.43	50	544.94

Table 7: Numbers of GMRES iterations and CPU times for Example 2 with the matching Schur complement preconditioner B_h.

$\texttt{tol} = 10^{-8}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M_1, M_2, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(16,16,16)	13	0.08	14	0.05	8	0.03	4	0.02	1	0.01
(32,32,32)	14	0.25	17	0.31	12	0.24	4	0.09	1	0.03
(64,64,64)	14	2.89	20	3.04	16	2.74	7	1.03	3	0.49
(128,128,128)	14	17.02	27	35.27	26	33.78	10	11.95	3	3.93
(256,256,256)	15	183.54	33	448.12	39	549.86	18	218.24	6	68.92

Table 8: Numbers of GMRES iterations and CPU times for Example 2 (fixing τ) with the matching Schur complement preconditioner B_h.

$\texttt{tol} = 10^{-8}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M_1, M_2, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(16,16,128)	14	0.39	26	0.73	26	0.68	10	0.27	3	0.11
(32,32,128)	14	1.03	26	2.15	26	2.02	10	0.73	3	0.25
(64,64,128)	14	4.14	26	8.34	26	8.24	10	2.91	3	1.00
(128,128,128)	14	17.28	27	35.56	26	33.85	10	11.95	3	3.98
(256,256,128)	15	85.09	27	168.87	26	159.72	10	54.14	3	18.22

Table 9: Numbers of GMRES iterations and CPU times for Example 2 (fixing h) with the matching Schur complement preconditioner B_h.

$\texttt{tol} = 10^{-8}$	$\gamma = 10^{-2}$		$\gamma = 10^{-4}$		$\gamma = 10^{-6}$		$\gamma = 10^{-8}$		$\gamma = 10^{-10}$	
(M_1, M_2, N)	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
(128,128,16)	13	2.02	14	2.09	8	1.17	4	0.61	1	0.23
(128,128,32)	14	4.24	18	5.61	12	3.61	4	1.25	1	0.46
(128,128,64)	14	8.43	20	12.75	17	10.61	7	4.21	3	1.99
(128,128,128)	14	17.39	27	36.20	26	34.24	10	11.98	3	3.99
(128,128,256)	14	34.77	33	93.30	38	112.56	18	45.52	6	14.64

system with the semismooth Newton (SSN) method [46], whereupon our preconditioned GMRES solver is employed to approximately solve the resulting linearized Jacobian system at each Newton iteration. The Jacobian matrices involved have a similar structure as the matrix (4) in the case without control constraints, with the exception that the $(2,2)$ block (given by $-\frac{1}{\gamma}\hat{I}_h$ in (4)) contains many very small diagonal terms relating to where in the domain Q the constraints are active. Our proposed preconditioner is therefore a good candidate for solving such amended systems. The only modification to our preconditioner is that we use the Schur complement pivoted about the $(1,1)$ block of the Jacobian matrix, since some diagonal entries of the $(2,2)$ block may become numerically zero due to active control constraints.

Both the outer SSN and inner GMRES methods are initialized with the zero vector. We denote by $\texttt{tol}_1 = 10^{-7}$ and $\texttt{tol}_2$ the stopping criteria for the outer SSN and the inner GMRES iterations, respectively.

Table 10: Numbers of GMRES iterations and CPU times for Example 2 (fixing $M_1 = M_2 = N = 64$) with a sequence of decreasing regularization parameter γ.

tol $= 10^{-14}$	Preconditioner P_h		Preconditioner B_h	
γ	Iter	CPU	Iter	CPU
10^{1}	5	0.47	11	1.65
10^{0}	6	0.56	11	1.65
10^{-1}	8	0.75	15	2.27
10^{-2}	10	0.94	43	7.49
10^{-3}	16	1.56	30	4.89
10^{-4}	33	3.70	38	6.41
10^{-5}	80	12.97	36	6.06
10^{-6}	179	47.47	33	5.44
10^{-7}	334	142.96	26	4.13
10^{-8}	504	305.03	18	2.75
10^{-9}	581	397.22	13	1.96
10^{-10}	628	457.76	10	1.48
10^{-11}	677	529.33	7	1.07
10^{-12}	730	608.32	5	0.78

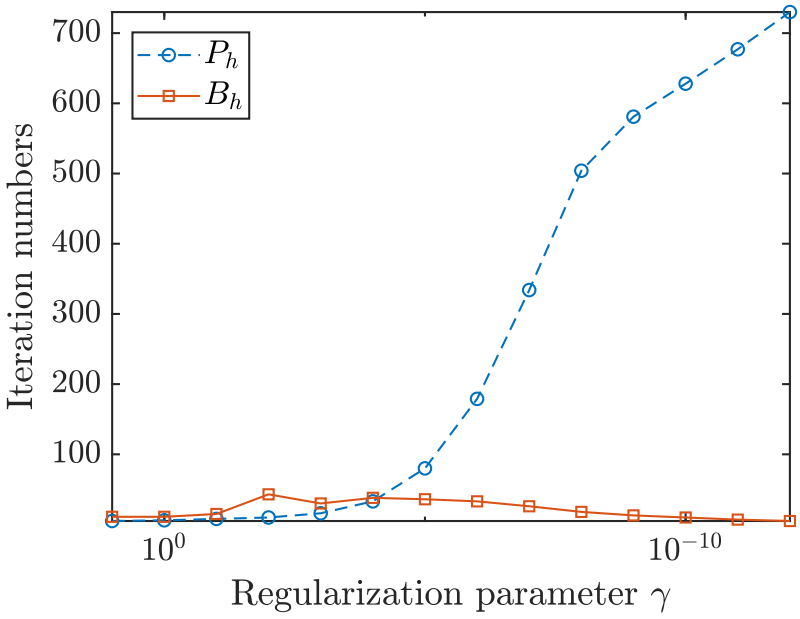

Figure 4: Comparison of iteration numbers with preconditioners P_h and B_h in Example 2 (with fixed $M_1 = M_2 = N = 64$).

Here, an alternative combination of the outer and inner tolerances may lead to better overall performance, but we did not further optimize our choice since we are mostly concerned with the performance of our proposed preconditioner for the inner GMRES iterations.

Let $\Omega = (0, 1)$ and $T = 2$. We choose $y_0(x) = \sin(\pi x)$, $y_1(x) = 0$, $u_a = 5$, $u_b = 10$,

$$f = -\max\{u_a, \min\{u_b, \sin(\pi x)(t-T)^2/\gamma\}\}, \text{ and } g(x,t) = 2\sin(\pi x) + \pi^2 \sin(\pi x)(t-T)^2 + \sin(\pi x)\cos(\pi t),$$

such that the exact solution is $y(x,t) = \sin(\pi x)\cos(\pi t)$ and $p(x,t) = \sin(\pi x)(t-T)^2$. The corresponding

optimal control can be derived from the projection formula $u = \max\{u_a, \min\{u_b, p/\gamma\}\}$.

Since the convergence results are largely similar to Example 1, to avoid duplication we only focus on the difference between the two preconditioners. Table 11 shows the number of iterations for both preconditioners by varying γ, while fixing h and τ, where we have used the same tolerance $\texttt{tol}_2 = 10^{-14}$ as in Table 5 for the inner GMRES iterations, so that we can easily compare them. The columns 'Iter1' and 'Iter2' denote the number of SSN iterations and the average number of preconditioned GMRES iterations, respectively. In the case where two-sided box constraints are included in the problem statement, the incorporation of these constraints seems to be beneficial for the convergence of the preconditioned GMRES solver, which was also observed in [21] with the preconditioner P_h. One possible explanation is that, for very small values of γ, the control constraints are active within the majority of the domain, which leads to the $(2,2)$ diagonal block of the Jacobian matrix being close to a zero matrix. Both preconditioners appear to perform very well with box constraints on the control, but in general this will not be the case. For instance, if we enforce only a one-sided control constraint (e.g., $u_a \leq u$), the iteration numbers with preconditioner P_h soar as γ gets smaller, whereas the iteration numbers with preconditioner B_h eventually decline in the case $M = N = 128$.

In Tables 12 and 13, we report the required number of iterations and CPU times using the preconditioner B_h, for varying h, τ, and γ. As expected, the outer SSN iterations demonstrate clear mesh-independent convergence. However, the numbers of GMRES iterations show very different behavior for the cases with box constraints and one-sided control constraints, respectively. In the case of one-sided control constraints, we do observe mildly increasing numbers of GMRES iterations as the mesh is refined and γ is decreased, as in the unconstrained cases. We believe this is caused by the $(2,2)$ block of the matrix system being highly ill-conditioned, due to the one-sided constraint being active in portions of the domain and inactive in others, which becomes more evident when the mesh-step size is sufficiently small to capture the abrupt changes. Figures 5 and 6 depict the computed optimal state, adjoint state, and optimal control (with its colormap) under the given control constraints, where the optimal control is reconstructed from the adjoint state. Finally, we highlight that the eigenvalue analysis based on the unconstrained case is not directly applicable to the constrained case, but numerically our proposed preconditioner B_h is effective for a range of examples. A rigorous justification of these observations would be an interesting topic of future work.

Table 11: Numbers of SSN iterations, average numbers of GMRES iterations, and total CPU times for Example 3 (fixing $M = N = 128$) with a sequence of decreasing regularization parameter γ.

	Box constraints: $u_a \leq u \leq u_b$						One-sided constraint: $u_a \leq u$					
$\texttt{tol}_2 = 10^{-14}$	Preconditioner P_h			Preconditioner B_h			Preconditioner P_h			Preconditioner B_h		
γ	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU
10^{1}	1	2.0	0.02	1	2.0	0.07	1	2.0	0.07	1	2.0	0.11
10^{0}	1	2.0	0.02	1	2.0	0.05	1	2.0	0.06	1	2.0	0.10
10^{-1}	4	5.8	0.15	4	19.3	1.28	4	7.0	0.34	4	26.5	1.76
10^{-2}	3	6.3	0.11	3	9.7	0.46	5	11.0	0.52	5	35.0	2.83
10^{-3}	3	4.7	0.08	3	9.0	0.42	4	16.3	0.66	4	53.8	3.88
10^{-4}	3	5.0	0.09	3	7.3	0.37	4	28.3	0.79	4	59.0	4.21
10^{-5}	2	3.5	0.04	2	5.5	0.22	4	48.0	1.35	4	54.8	3.73
10^{-6}	2	2.5	0.03	2	3.0	0.11	4	84.5	2.62	4	41.8	2.81
10^{-7}	2	4.0	0.05	2	2.0	0.08	4	150.3	6.14	4	56.5	4.07
10^{-8}	2	4.0	0.05	2	2.0	0.09	4	250.5	14.22	4	62.8	4.68
10^{-9}	2	4.0	0.04	2	2.0	0.08	4	356.5	26.28	4	55.8	4.16
10^{-10}	2	4.0	0.04	2	2.0	0.09	6	859.3	208.02	4	33.5	2.46
10^{-11}	2	4.0	0.05	2	2.0	0.07	8	1118.6	386.63	4	25.0	1.78
10^{-12}	2	4.0	0.05	2	2.0	0.08	9	1210.3	486.43	4	34.8	2.62

Table 12: Numbers of SSN iterations, average numbers of GMRES iterations, and total CPU times for Example 3 (box constraints: $u_a \le u \le u_b$) with the matching Schur complement preconditioner B_h.

$\texttt{tol}_2 = 10^{-10}$	$\gamma = 10^{-2}$			$\gamma = 10^{-4}$			$\gamma = 10^{-6}$			$\gamma = 10^{-8}$		
(M, N)	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU
(128,128)	3	7.7	0.4	3	6.0	0.3	2	2.5	0.1	2	2.0	0.1
(256,256)	3	7.7	1.0	3	6.0	0.8	2	4.0	0.4	2	2.0	0.2
(512,512)	3	8.3	3.7	3	6.3	2.9	2	4.0	1.3	2	2.5	0.9
(1024,1024)	3	8.3	13.2	3	6.3	10.1	2	4.5	5.1	2	3.5	4.1
(2048,2048)	3	8.3	51.9	3	6.3	38.8	2	4.5	19.3	2	3.5	15.7

Table 13: Numbers of SSN iterations, average numbers of GMRES iterations, and total CPU times for Example 3 (one-sided constraint: $u_a \le u$) with the matching Schur complement preconditioner B_h.

$\texttt{tol}_2 = 10^{-10}$	$\gamma = 10^{-2}$			$\gamma = 10^{-4}$			$\gamma = 10^{-6}$			$\gamma = 10^{-8}$		
(M, N)	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU	Iter1	Iter2	CPU
(128,128)	5	16.4	1.3	4	27.8	1.7	4	31.8	1.9	4	31.8	2.0
(256,256)	5	17.0	3.8	4	31.3	6.2	4	46.0	10.3	4	43.5	10.7
(512,512)	5	17.6	13.5	4	33.5	23.6	4	63.8	56.3	4	61.3	59.7
(1024,1024)	4	17.3	38.6	4	35.0	93.7	4	79.0	304.3	4	85.3	386.0

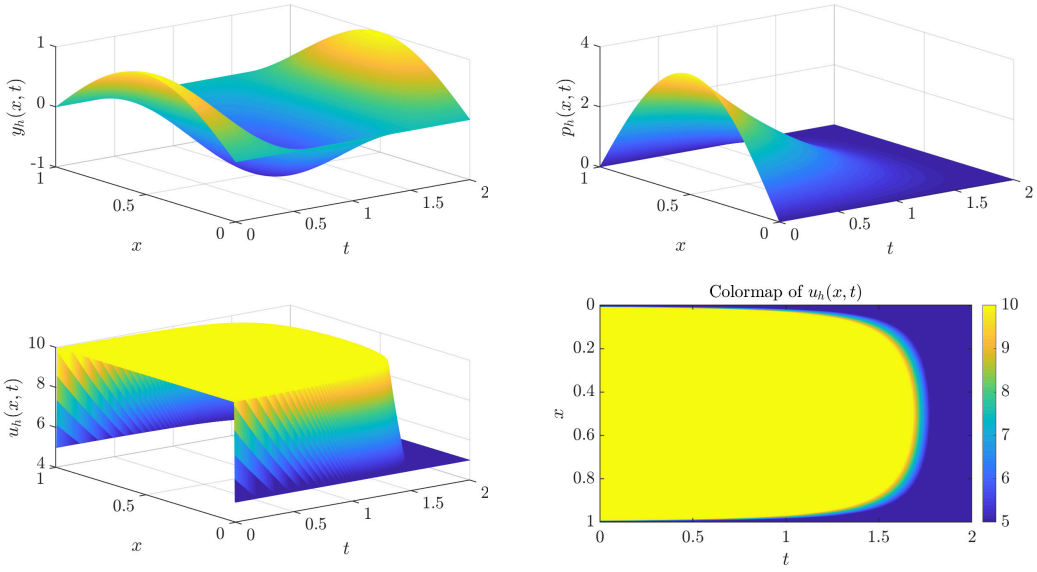

Figure 5: Computed y_h, p_h, and control $u_h = \max\{5, \min\{10, p_h/\gamma\}\}$ with box constraints, with $\gamma = 10^{-2}$ in Example 3 ($M = N = 1024$).

5. Conclusion

In this paper, we have proposed and analyzed a matching Schur complement preconditioner for iteratively solving the finite-difference discretized systems arising from optimal control problems involving wave equations. The solver we have developed could also be transferred to matrix systems obtained from finite element methods. Numerical results from both 1D and 2D examples are shown to confirm our theoretical

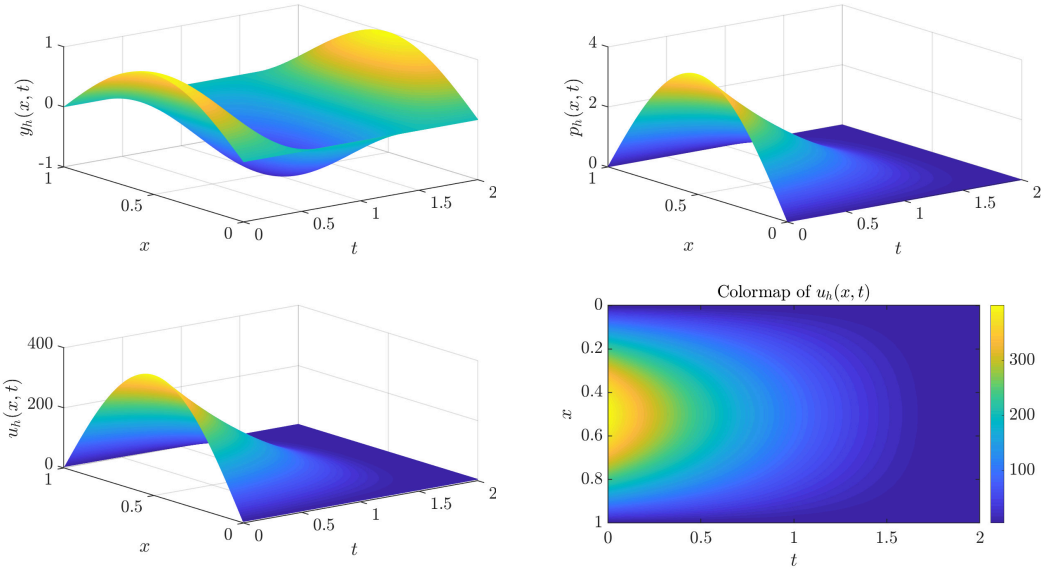

Figure 6: Computed y_h, p_h, and control $u_h = \max\{5, p_h/\gamma\}$ with one-sided constraint, with $\gamma = 10^{-2}$ in Example 3 ($M = N = 1024$).

analysis. In contrast to the established constraint preconditioner, our new matching Schur complement preconditioner shows significantly faster convergence rates when the regularization parameter γ becomes very small, while also achieving convergence rates independent of the step-size h. It remains an open problem to design a preconditioner that achieves a convergence rate independent of both mesh step-sizes (h and τ) and the regularization parameter γ for the linear system under consideration, although based on the numerical challenges that result from solving similar forward problems it is far from guaranteed that this is even achievable.

In the future, it would be of interest to design a hybrid preconditioner (based on a switching value of γ) that combines the advantages of both the constraint preconditioner P_h and the matching Schur complement preconditioner B_h. Furthermore, the h-independent and γ-robust behavior of the new preconditioner B_h may open the door to applying solvers of this form alongside schemes that improve the performance in the time variable, such as multiple shooting methods [18], parareal schemes [47, 48], or instantaneous control [49]. Finally, modifications to the problem formulation could be made, such as introducing different norms within the cost functional, or considering nonlinear wave equations and boundary control problems; we believe that more sophisticated problem formulations such as these could also be tackled using iterative solvers of the type described in this paper.

Acknowledgments. The authors would like to thank an anonymous referee for their constructive and valuable comments.

Appendix A

Lemma A.1. *Let $Z_N = pendiag([1, -2, 1, 0, 0])$ be a square matrix with dimension $N \times N$, then its inverse is a lower triangular Toeplitz matrix with the following expression:*

$$Z_N^{-1} = \text{LTToeplitz}([1, 2, \cdots, N-1, N]^\intercal),$$

in which $[1, 2, \cdots, N-1, N]^\intercal$ *gives the first column of* Z_N^{-1}.

Proof. We prove the above statement by mathematical induction on the dimension N. When $N = 3$, the result holds, since it is straightforward to verify that

$$Z_3^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} = \text{LTToeplitz}([1, 2, 3]^\intercal).$$

Now, we assume the conclusion is true for $N = k$ and proceed to show the conclusion also holds for $N = k+1$. When $N = k + 1$, we have the following recursive block structure:

$$Z_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & -2 & 1 & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ z & Z_k \end{bmatrix},$$

where $z = [-2, 1, 0, \cdots, 0]^\intercal$. A simple step of block Gaussian elimination leads to

$$\begin{bmatrix} 1 & 0 \\ -z & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z & Z_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & Z_k \end{bmatrix},$$

which hence implies

$$Z_{k+1}^{-1} = \begin{bmatrix} 1 & 0 \\ z & Z_k \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & Z_k^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -z & I \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -Z_k^{-1} z & Z_k^{-1} \end{bmatrix}. \tag{A.1}$$

By the inductive assumption it holds that

$$Z_k^{-1} = \text{LTToeplitz}([1, 2, \cdots, k-1, k]^\intercal).$$

Based on this, we can also easily obtain (noting that $z = [-2, 1, 0, \cdots, 0]^\intercal$ only contains non-zeros in the first two entries) that

$$-Z_k^{-1} z = [2, 3, \cdots, k+1]^\intercal.$$

Inserting both blocks back into (A.1), we arrive at the following desired lower triangular Toeplitz matrix:

$$Z_{k+1}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & 0 & \cdots & 0 \\ 3 & 2 & 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ k & \ddots & \ddots & 2 & 1 & 0 \\ k+1 & k & \cdots & 3 & 2 & 1 \end{bmatrix} = \text{LTToeplitz}([1, 2, \cdots, k-1, k, k+1]^\intercal),$$

which therefore completes the proof by the principle of mathematical induction. □

References

[1] A. Borzì, K. Kunisch, D. Y. Kwak, Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system, SIAM J. Control Optim. 41 (5) (2003) 1477–1497.

[2] A. Borzì, V. Schulz, Computational Optimization of Systems Governed by Partial Differential Equations, SIAM, Philadelphia, PA, 2012.

[3] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Springer, New York, NY, 2009.

[4] J.-L. Lions, Optimal Control of Systems Governed by Partial Differential Equations, Springer–Verlag, New York, NY, 1971.

[5] F. Tröltzsch, Optimal Control of Partial Differential Equations, AMS, Providence, RI, 2010.

[6] M. D. Gunzburger, Perspectives in Flow Control and Optimization, SIAM, Philadelphia, PA, 2003.

[7] A. Jameson, Aerodynamic design via control theory, J. Sci. Comput. 3 (3) (1988) 233–260.

[8] M. Bergounioux, X. Bonnefond, T. Haberkorn, Y. Privat, An optimal control problem in photoacoustic tomography, Math. Models Methods Appl. Sci. 24 (12) (2014) 2525–2548.

[9] J. W. Pearson, A. J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numer. Linear Alg. Appl. 19 (5) (2012) 816–829.

[10] J. W. Pearson, A. J. Wathen, Fast iterative solvers for convection–diffusion control problems, Electron. Trans. Numer. Anal. 40 (2013) 294–310.

[11] J. W. Pearson, M. Stoll, A. J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, SIAM J. Matrix Anal. Appl. 33 (4) (2012) 1126–1152.

[12] J. W. Pearson, M. Stoll, Fast iterative solution of reaction–diffusion control problems arising from chemical processes, SIAM J. Sci. Comput. 35 (5) (2013) B987–B1009.

[13] J. Schöberl, W. Zulehner, Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems, SIAM J. Matrix Anal. Appl. 29 (3) (2007) 752–773.

[14] J. Schöberl, R. Simon, W. Zulehner, A robust multigrid method for elliptic optimal control problems, SIAM J. Numer. Anal. 49 (4) (2011) 1482–1503.

[15] W. Zulehner, Nonstandard norms and robust estimates for saddle point problems, SIAM J. Matrix Anal. Appl. 32 (2) (2011) 536–560.

[16] Z.-Z. Bai, Block preconditioners for elliptic PDE-constrained optimization problems, Computing 91 (4) (2010) 379–395.

[17] Z.-Z. Bai, M. Benzi, F. Chen, Z.-Q. Wang, Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, IMA J. Numer. Anal. 33 (1) (2012) 343–369.

[18] M. Heinkenschloss, A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems, J. Comput. Appl. Math. 173 (1) (2005) 169–198.

[19] A. Rincon, I.-S. Liu, On numerical approximation of an optimal control problem in linear elasticity, Divulg. Mat. 11 (2) (2003) 91–107.

[20] M. Bergounioux, K. Ito, K. Kunisch, Primal-dual strategy for constrained optimal control problems, SIAM J. Control Optim. 37 (4) (1999) 1176–1194.

[21] B. Li, J. Liu, M. Xiao, A fast and stable preconditioned iterative method for optimal control problem of wave equations, SIAM J. Sci. Comput. 37 (6) (2015) A2508–A2534.

[22] M. Porcelli, V. Simoncini, M. Tani, Preconditioning of active-set Newton methods for PDE-constrained optimal control problems, SIAM J. Sci. Comput. 37 (5) (2015) S472–S502.

[23] A. Schiela, S. Ulbrich, Operator preconditioning for a class of inequality constrained optimal control problems, SIAM J. Optim. 24 (1) (2014) 435–466.

[24] M. Stoll, A. Wathen, Preconditioning for partial differential equation constrained optimization with control constraints, Numer. Linear Alg. Appl. 19 (1) (2012) 53–71.

[25] O. L. Elvetun, B. F. Nielsen, PDE-constrained optimization with local control and boundary observations: Robust preconditioners, SIAM J. Sci. Comput. 38 (6) (2016) A3461–A3491.

[26] K.-A. Mardal, B. F. Nielsen, M. Nordaas, Robust preconditioners for PDE-constrained optimization with limited observations, BIT Numer. Math. 57 (2) (2017) 405–431.

[27] J. W. Pearson, J. Gondzio, Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization, Numer. Math. 137 (4) (2017) 959–999.

[28] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.

[29] K. Chen, Matrix Preconditioning Techniques and Applications, Cambridge University Press, Cambridge, 2005.

[30] Y. Saad, Iterative Methods for Sparse Linear Systems: Second Edition, SIAM, Philadelphia, PA, 2003.

[31] J. W. Pearson, A. Wathen, Matching Schur complement approximations for certain saddle-point systems, in: Contemporary Computational Mathematics – A Celebration of the 80th Birthday of Ian Sloan, Editors: J. Dick, F. Y. Kuo, H. Wozniakowski, Springer, 2018.

[32] C. C. Paige, M. A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (4) (1975) 617–629.

[33] W. Hackbusch, Elliptic Differential Equations: Theory and Numerical Treatment, Springer Berlin Heidelberg, 2017.

[34] O. G. Ernst, Residual-minimizing Krylov subspace methods for stabilized discretizations of convection–diffusion equations, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1079–1101.

[35] J. Liesen, Z. Strakos, Convergence of GMRES for tridiagonal Toeplitz matrices, SIAM J. Matrix Anal. Appl. 26 (1) (2004) 233–251.

[36] S. Noschese, L. Pasquini, L. Reichel, Tridiagonal Toeplitz matrices: properties and novel applications, Numer. Linear Alg. Appl. 20 (2) (2012) 302–326.

[37] G. D. Smith, Numerical Solution of Partial Differential Equations: Finite Difference Methods (3rd Edition), Clarendon Press, Oxford, 1985.

[38] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric matrix systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.

[39] Z.-Z. Bai, Structured preconditioners for nonsingular matrices of block two-by-two structures, Math. Comp. 75 (254) (2005) 791–816.

[40] W. Govaerts, J. Pryce, A singular value inequality for block matrices, Linear Alg. Appl. 125 (1989) 141–148.

[41] M. F. Murphy, G. H. Golub, A. J. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (6) (2000) 1969–1972.

[42] Y. Notay, A new analysis of block preconditioners for saddle point problems, SIAM J. Matrix Anal. Appl. 35 (1) (2014) 143–173.

[43] D. Silvester, H. Elman, A. Ramage, Incompressible Flow and Iterative Solver Software (IFISS) version 3.6, `http://www.manchester.ac.uk/ifiss/` (February 2019).

[44] H. Elman, A. Ramage, D. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow, ACM Trans. Math. Softw. 33 (2007) 2–14.

[45] H. Elman, A. Ramage, D. Silvester, IFISS: A computational laboratory for investigating incompressible flow problems, SIAM Rev. 56 (2014) 261–273.

[46] A. Kröner, K. Kunisch, B. Vexler, Semismooth Newton methods for optimal control of the wave equation with control constraints, SIAM J. Control Optim. 49 (2) (2011) 830–858.

[47] Y. Maday, G. Turinici, A parareal in time procedure for the control of partial differential equations, C. R. Math. 335 (4) (2002) 387–392.

[48] T. P. Mathew, M. Sarkis, C. E. Schaerer, Analysis of block parareal preconditioners for parabolic optimal control problems, SIAM J. Sci. Comput. 32 (3) (2010) 1180–1200.

[49] H. Choi, M. Hinze, K. Kunisch, Instantaneous control of backward-facing step flows, Appl. Numer. Math. 31 (2) (1999) 133–158.